electronics

MDPI

Article

Upsampling Real-Time, Low-Resolution CCTV Videos Using Generative Adversarial Networks

Debapriya Hazra and Yung-Cheol Byun *

Department of Computer Engineering, Jeju National University, 102 Jejudaehak-ro, Jeju-si 63243, Korea; debapriyah@gmail.com

* Correspondence: ycb@jejunu.ac.kr

Received: 15 July 2020; Accepted: 11 August 2020; Published: 14 August 2020

Abstract: Video super-resolution has become an emerging topic in the field of machine learning. The generative adversarial network is a framework that is widely used to develop solutions for low-resolution videos. Video surveillance using closed-circuit television (CCTV) is significant in every field, all over the world. A common problem with CCTV videos is sudden video loss or poor quality. In this paper, we propose a generative adversarial network that implements spatio-temporal generators and discriminators to enhance real-time low-resolution CCTV videos to high-resolution. The proposed model considers both foreground and background motion of a CCTV video and effectively models the spatial and temporal consistency from low-resolution video frames to generate high-resolution videos. Quantitative and qualitative experiments on benchmark datasets, including Kinetics-700, UCF101, HMDB51 and IITH_Helmet2, showed that our model outperforms the existing GAN models for video super-resolution.

Keywords: closed-circuit television (CCTV); generative adversarial networks (GAN); spatio-temporal network; low-resolution videos; high-resolution videos

1. Introduction

With progress in technology, the mass surveillance industry has a constant need to enhance the security analysis mechanism. One of the major resources for video surveillance is closed-circuit television (CCTV). CCTV cameras are widely used to identify criminal activities, monitor activities for evidence collection and for many other security purposes. The British transport police recorded that in 45% of the cases in which CCTV was available, 29% of the cases could be solved due to the presence of CCTV [1].

There is an enormous amount of video content generated in these CCTV videos. It is essential to compress these video streams, to decrease the mandatory bandwidth and storage. CCTV in video surveillance is a popular technology, used in various areas, such as public transport, security in various places, e.g., schools, hospitals and police stations, fact detection, etc. These technologies are maturing rapidly. According to [2], many applications related to CCTV and video surveillance ranked higher in 2018 compared to 2015. A few examples include left-luggage detection, overcrowding detection and rail track access detection. This shows that there has been a subtle yet positive shift from 2015–2018 in the public perception of CCTV and surveillance technologies. The perception has had a meaningful shift from security aspects toward the enhancement of the actual security and public's experience in different areas. As per [3], the global video surveillance market is projected to earn 74.6 billion USD in revenue, and today global video surveillance and CCTV is nearly three times as prominent as in 2016.

With such enormous amounts of video and CCTV data produced every day, it is important to enhance the quality of footage so that the video quality can be improved. Therefore, we aim to upsample CCTV videos, specifically focusing on videos with low resolution. One such example

includes videos recorded at night time. Their quality is usually not up to the mark for reasons such as dim or no light. The main contribution of this paper is to solve the real-time low-resolution CCTV video issue and generate high-resolution videos that are productive for further usage in case of any emergency. We have used generative adversarial networks to achieve our objective.

In our work we have proposed a novel model with the following contributions:

1. Dual spatio-temporal generators and discriminators to enhance the low-resolution CCTV videos with high accuracy.
2. Concatenated reversible residual blocks in generators which help in distinguishing between low and high-resolution frames and extract intricate features without information loss.
3. Mapping between low and high-resolution frames in an adaptive way by using a sparse matrix for fused features.
4. Discriminators with R(2 + 1)D convolutional ResNet for better optimization and performance.

The generator's architecture consists of the spatial generator G_S and the temporal generator G_T. Consecutive low-resolution frames are provided to the generator G_S which generates the initial feature maps. We have used multiple reversible residual blocks (RRB) in parallel and concatenated the output feature maps generated from every block in each level to the next parallel level to learn the difference between low-resolution and high-resolution frames. The concatenated feature maps are then fed to the temporal generator G_T which outputs the feature maps of continuous frames. A sparse matrix is used to fuse the features, after which a reconstruction layer is used that produces high-resolution frames.

The spatio-temporal discriminators were provided with real data and samples from the generator. It got individual frames as input and uses a ResNet architecture to determine whether a frame was generated from a real video clip or from the generator. In contrast, consecutive frames were provided to the temporal discriminator and trained using a (2 + 1)D convolution ResNet that critiqued real and generated videos. The generator was trained to fool both the discriminators.

The results show that our approach requires a lesser computational overhead and produces a better outcome than existing GAN models for video super resolution. Our model was trained on different large scale datasets and we were able to extract different latent features while focusing on both the foreground and background footage.

The rest of this paper includes related works in Section 2, an explanation of the datasets used for experiments in Section 3, the proposed methodology in Section 4, environment details in Section 5 and the results and performance evaluation of the model in Section 6. Section 7 concludes this article.

2. Related Works

There has been a remarkable evolution of innovative technologies in the field of image processing, such as encryption [4], masking [5] and segmentation [6], which helps us to identify small changes in an image. High-quality generative models have been developed to create natural images using robust matrices. Clark et al. [7] recommended using a capable machine learning-based solution dual video discriminator GAN, for long and high-definition videos. They evaluated actions related to video synthesis and video prediction, and attained good results. Videos obtained by CCTV cameras play a vital role in crime prevention [8]. They solved the problem of natural video modeling by introducing GAN, which can handle the complexity of sizeable natural video datasets. This has been proven using UCF-101 and frame-conditional Kinetics-600 datasets, which have high-quality video and diversity. Rigid data standards set by the DVD-GAN were used as a reference point for the previous generation modeling community. Yang et al. [9] introduced a new high-definition video with a method that reduces the depth of the frame and the background of the integrated deep network's inter-frame movement. Unlike traditional methods, the proposed residual space network studies local and terrestrial remains, including high-resolution frames and similar low-resolution frames. It includes frame and adjacent high-resolution frame differences, which then use the video sequence in a circular circuit network

to randomly connect these frames and interframes and predict the direction of high-resolution time remnants in the second layer, to estimate the video's super-resolution. However, their approach does not fit in with long-term video storage, in order to rebuild a high-resolution framework.

Ballas et al. [10] provided a way to learn the unique and temporary attributes known as "percepts" by gated-recurrent-unit recurrent networks. This method was based on the information derived from all the deep networks trained in the Big Image Network database. Although there is discriminatory information about higher levels of perception, the distance is small, but lower levels of awareness can maintain better historical accuracy and mimic different behavior patterns. Usage of low-level knowledge can lead to higher video formats, reduce this effect and control several parameters. GRU models that use repetition functions to achieve smaller relationships with input space model units set out to empirically demonstrate how a person can be identified by recorded human behavior. Especially on the YouTube2Text dataset, they got an identical effect as a simple decoding model in the implementation of 3D-CNN.

In their paper, Tran et al. [11] discussed various forms of spatiotemporal convolutions and studied their effects on the learning process. According to their research, stimulus is a two-dimensional CNN network implemented in a video frame. In their study, they demonstrated the specific advantages of a 3D-CNN network over a two-dimensional CNN network in the rest of the residual learning environment. Combining 3D convolutional filters with different spatial and temporal settings also shows that accuracy can be significantly improved. Experiments show that CNN's "R(2 + 1)D" spatiotemporal convolutional block is comparable to Sport-1M, Kinetics, UCF101 and HMDB51 datasets.

Video clips can be categorized by content and behavior. Content describes inclusion in the video, and process defines the dynamics. Tulyakov et al. [12] suggested motion and content decomposed generative adversarial network for video content. The proposed frame produces a random movement vectors of the entire sequence. Some structured weaknesses have specific features and steps. Some things are always revealed, but some actions seem unplanned. To learn the power of motion and content, they introduced a powerful new video and video advertising program. Complete test results across a wide range of complex datasets, including performance and value from today's viewing environment, confirm the proposed framework's effectiveness. Besides,the motion and content decomposed generative adversarial network explains that one can create videos with a variety of themes.

Network video training of the generative adversarial network is difficult due to the size and complexity of each dataset. The resolution usually measures all GAN rates. Saito et al. [13] provided an effective method to learn the actual storage of high-definition video datasets through non-practical learning. Its computational cost was limited due to this decision. They achieved this by designing the generator model as a bunch of small sub-generators and training the model in a specific way. They trained each sub-generator in a unique style. During the training process, they always introduced auxiliary sub-mapping layers between each sub-generator, thereby reducing the frame rate by a certain percentage. The proposed system allows each sub-generator to customize different video qualities. Additionally, only a small GPU is required to train a high processor for stability.

In their paper, Saito et al. [14] suggested the creation of temporal generative adversarial nets to analyze unlabeled video inputs and video segments. By combining a video production system with a single 3D generator for production, the system uses two methods: short-term production and reflection. Entries have a set of different settings that are suitable for each photo and video. Video generators create a digital video system with latent variables. In these new network connections, they developed and trained the types of products needed to address GAN training's weaknesses, such as the Wasserstein GAN.

Despite recent advances in digital photography, models that accurately represent important decisions move away from multiple platforms such as ImageNet. To do this, Brock et al. [15] thoroughly tested the generative adversarial network's trainers and learned independent management at this level. They found that using orthogonal adjustments on players simplified by the "truncation trick", could

better handle the difference between model brightness and contrast by reducing output-input diversity. Changes in model products define new forms of art, but there are very few adjustments. ImageNet dataset training with 128 × 128 resolution achieved 166.5 inception Score and Frechet inception of 7.4.

Residual networks perform the most advanced image processing to enhance and significantly improve the system. However, Gomez et al. [16] used traverse multiplication to calculate the division using memory. They introduced ResNet's variable, the reversible residual network, to fully reactivate each layer in the next layer. Therefore, most of the layer activations need not be stored in memory during bag augmentation. CIFAR-10, CIFAR-100 and ImageNet show RevNet's results are better. Even if active storage requirements are not related to depth, it can create almost the same sorting accuracy as a ResNet of the same size. However, their proposed system does not work better on a more significant and robust network with limited computer resources.

The reality of the enhancing video resolution is that it is a challenging job that inspires public interest in research and industry. Zhu et al. [17] suggested a residual invertible spatio-temporal network, a completely new architecture with ultra-high-density video. The residual invertible spatio-temporal system can properly use necessary information from low resolution to current resolution and temporarily edit video frames from standard frames. The residual invertible spatio-temporal network is deeper and more potent than current repetitive systems. The quality of light response of spatial components is designed to minimize information loss during job conversion and ensure the consistency of job features. This volatile component provides new recurring coordinates with multiple tooth connections, thereby not deepening the network and not compromising on functionality. As part of the reconstruction component, an original fusion method was proposed based on a sparse technique that combines spatial and temporal properties. According to experiments using universal quality data packets, the residual invertible spatio-temporal network improves on the latest methods. Moreover, authors in [18], proved the importance of choosing a proper activation function for the hidden layers using a mathematical evaluation that helped the model to work on complex mappings required for vast and non-linear data. Hongwei Guo et al. in [19] also utilized the advantages of backpropagation algorithms and mathematically induced one for thin plate bending problems. In our proposed work we have considered the factors mentioned in [18,19] for better performance.

The generative model of images has evolved into high-resolution samples using robust scales. Clark et al. [7] aimed to prove success in video modeling by demonstrating that large-scale obstetric antagonistic networks trained on the complex Kinetics-600 dataset can produce video samples that are more complex than previous works. The dual video discriminator GAN model uses practical discrimination calculations to span longer, higher resolution videos. They evaluated tasks related to video collection and video prediction and set the latest Fréchet inception distance for predicting Kinetics-600 and the latest starting point for collecting UCF-101 datasets.

Vondrick et al. [20] used several unlabeled videos to learn the visual and dynamic patterns of video recognition tasks and video production tasks. They proposed a generative adversarial network for video creation with a spatio-temporal convolutional structure that incorporates the scene's background and foreground. According to experiments, the model can create a small video at the full frame rate, which is superior to simple metrics, can reach one second per total frame rate, and is also useful for predicting a reasonable future for still images. Besides, the results of experiments and visualizations indicate that the model has learned useful functions internally and can recognize the procedure with minimal supervision. Hence, scene dynamics are a promising learning signal. Creating video models can affect many video understanding and simulation applications.

In the past few years, the process of machine learning has accelerated. The super-resolution convolutional neural network (SRCNN) model by Dong et al. [21] integrated convolutional neural networks (CNN) and ISRR, and they developed a network architecture for building training strategies [22–25]. However, these methods give outstanding results without elaborating on the maximum frequency. It has been recommended by Johnson et al. [26] to compute the perceptual loss of an ultra-resolution model, not the pixels in operational space. The generative adversary network

(GAN) [27] was introduced by [23,28] to challenge this network to create more original and creative content. Lim et al. [29] removed the batch normalization layer of the generative adversarial network for image super-resolution to draw the deep-registration dual network. Xintao Wang et al. [30] developed the generator network using residual density blocks. Unfortunately, the effectiveness of image reconstruction has been improved, but on the contrary, these methods still have some negative artifacts.

3. Data

In our proposed work, we trained and evaluated our model qualitatively and quantitatively using several datasets, as mentioned in the following subsections. We trained our model to upsample any low-resolution videos and provided results in the evaluation section for 512 pixels, 720 pixels and 960 pixels. For training we downscaled the input videos which are mentioned as low-resolution videos, and for output we performed upscaling to high-resolution videos.

3.1. Kinetics-700

We trained our model on the large Kinetics-700 dataset [31]. The Kinetics-700 dataset has a collection of 650 k video clips of human-object interactions each lasting around 10 s. The clips are YouTube video clips of diverse frame rate and resolution that help in training large models without causing worry about overfitting, as in the case of small fixed attribute datasets. The dataset consists of 700 human action classes with each class having at least 900 video clips. There is a standard validation set, the Kinetics-700 dataset, with 540 k video clips corresponding to training set.

3.2. UCF101

UCF101 is an action recognition dataset consisting of 101 action classes [32]. The UCF101 data has 13,320 videos from YouTube with various camera motions, backgrounds, poses, objects and visual effects. We split the data into 70% for training and 30% for testing. The 101 categories were subdivided into 25 categories. Each category had a maximum of seven videos of a particular action with some common features.

3.3. HMDB51

The HMDB51 dataset is a human motion dataset containing 2 GB of video data with total of 7000 video clips [33]. The clips are assigned to 51 action classes. HMDB51 data were collected from different sources, such as Google videos, YouTube, Prelinger archive and movies. The dataset is categorized into "no motion" and "camera motion" and provides an overall coverage of the video from different angles of the camera. Video quality in the HMDB51 dataset is graded on 3 levels. "Good" quality videos are considered to be the one wherein each finger of the human in the video can be identified and "medium" or "bad" quality videos are the ones where any one or more body parts cannot be recognized while the video is being played. For HMDB51 dataset also, we divided the data into 70% and 30% for training and testing purposes.

3.4. IITH_Helmet2

IITH_Helmet2 is a dataset consisting of videos from crowded traffic from Hyderabad city CCTV network in India [34]. It consists of 15 GB of video data collected at 30 frames per second. Since these dataset consists of real-time CCTV videos, we split it into 50% for training and 50% for evaluation.

4. Proposed Generative Adversarial Networks for Upsampling Real-Time Low-Resolution CCTV Videos

4.1. Overview

Ian Goodfellow, along with his colleagues, designed a machine learning framework named the generative adversarial network (GAN) in the year 2014 [27]. Since then, the GAN has been proved to be of tremendous use in various field for developing different applications with high efficiency. The GAN framework trains two models: the generative model G and the discriminative model D. G captures the data distribution p over data x from the random input noise variable $p_n(n)$. The generator data are fed into the discriminator along with the real input ground-truth data. The discriminator then evaluates the probability of whether x came from p or from the real data distribution. During training G tries to fool the discriminator D, whereas D tries to correctly label real and generated data. Thus G strives to minimize $log(1 - D(G(n)))$ and both the models G and D compete against each other in a minmax game where the value function $F(D, G)$ is defined as (1) [27]:

$$\min_G \max_D F(D,G) = \mathbb{E}_{x\sim p(x)}[logD(x)] + \mathbb{E}_{n\sim p_n(n)}\,[log(1 - D(G(n)))] \quad (1)$$

CCTV videos are some of the most crucial proof for any security investigation, but they mostly contain a lot of noise, especially in the dark, and are often blurry and non-productive. Our main objective in this study was to enhance the quality of real-time CCTV videos by applying generative adversarial networks. Figure 1 shows the architecture flow of basic GAN model and our proposed GAN model.

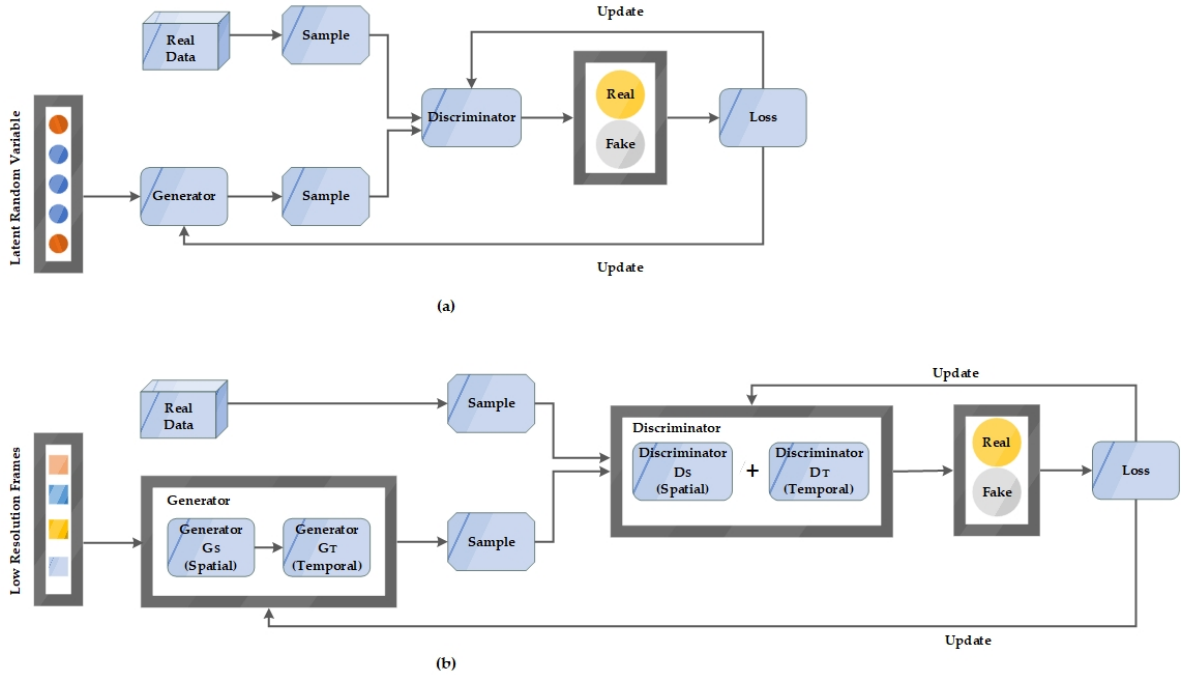

Figure 1. Basic flow of (**a**) the GAN model and (**b**) the proposed GAN model.

In our proposed work, we used a dual generator and dual discriminator to upsample the low-resolution CCTV videos. Inspired from RISTN [17], we built our generator architecture consisting of the spatial generator G_S and the temporal generator G_T. The generator G_S is provided with consecutive low-resolution frames which are used to generate initial feature maps using zero padding on RGB channels. We use multiple reversible residual blocks (RRB) in parallel and concatenated the output feature maps generated from every block in each level to the next parallel level. This way, the reversible blocks learns to differentiate between the low-resolution and high-resolution

frames. The concatenated feature maps are then fed to the temporal generator G_T which outputs the feature maps of continuous frames. The outputs or the features from both G_T and G_S are fused using a sparse matrix, after which we use a deconvolutional layer which retrieves every pixel and produces high-resolution frames.

The discriminator is a combination of a spatial (D_S) and a temporal discriminator (D_T), which work separately. The combination of two discriminators is defined with a "+" sign in Figure 1. D_S and D_T are provided with the real data distribution and the sample from the generator. The spatial discriminator gets individual frames as input and then uses a ResNet architecture to determine whether a frame is generated from a real video clip or from the generator. Contrastingly, D_T, the temporal discriminator, is trained on consecutive frames using a (2 + 1)D convolution ResNet and tries to discover whether the video was sampled from real or generated dataset. The generator is trained to fool both the discriminators.

4.2. Spatial Generator

It is important to retain the spatial information of the videos. The spatial information assures that the low-resolution frames and the generated high-resolution frames have maximum structural similarity. We achieved this by building the spatial generator G_S using reversible residual blocks (RRB). The *RRBs* helps in retrieving and reconstructing the spatial features without any loss. The *RRBs* are constructed in parallel and the output feature maps from every block at each level is concatenated and sent to the next level, which trains them by comparing and learning the differences between low-resolution and high-resolution frames. The RRBs are shown in Figure 2 where every RRB consists of a forward and a reverse computation. F and R are the forward and reverse residual functions. These functions are composed of convolutions, batch normalization layers and a rectified linear unit with a residual block having stride 1 to preserve information and avoid loss. In Figure 2 $a1$ and $a2$ can be considered as input features and $b1$ and $b2$ are the output features produced from the additive coupling rule [35,36]. The output features $b1$ and $b2$ can be computed as in (2). These reversible residual blocks reduce the memory footprint which helps us to work with large video datasets without any degradation in performance.

$$b1 = a1 + R(a2) \quad \text{and} \quad b2 = a2 + F(a1) \tag{2}$$

Activation of each layer can be reformed from the next layer's activation with the following Equation (3):

$$a1 = b1 - R(b2) \quad \text{and} \quad a2 = b2 - F(b1) \tag{3}$$

From the above equations (Equations (2) and (3)), we can infer that given the features of the nth layer, we can compute the features of the previous layer. *RRBs* are memory efficient and the continuous processing of the concatenated feature maps to the next level makes the network efficient enough to estimate the difference between the low-resolution and targeted high-resolution feature maps.

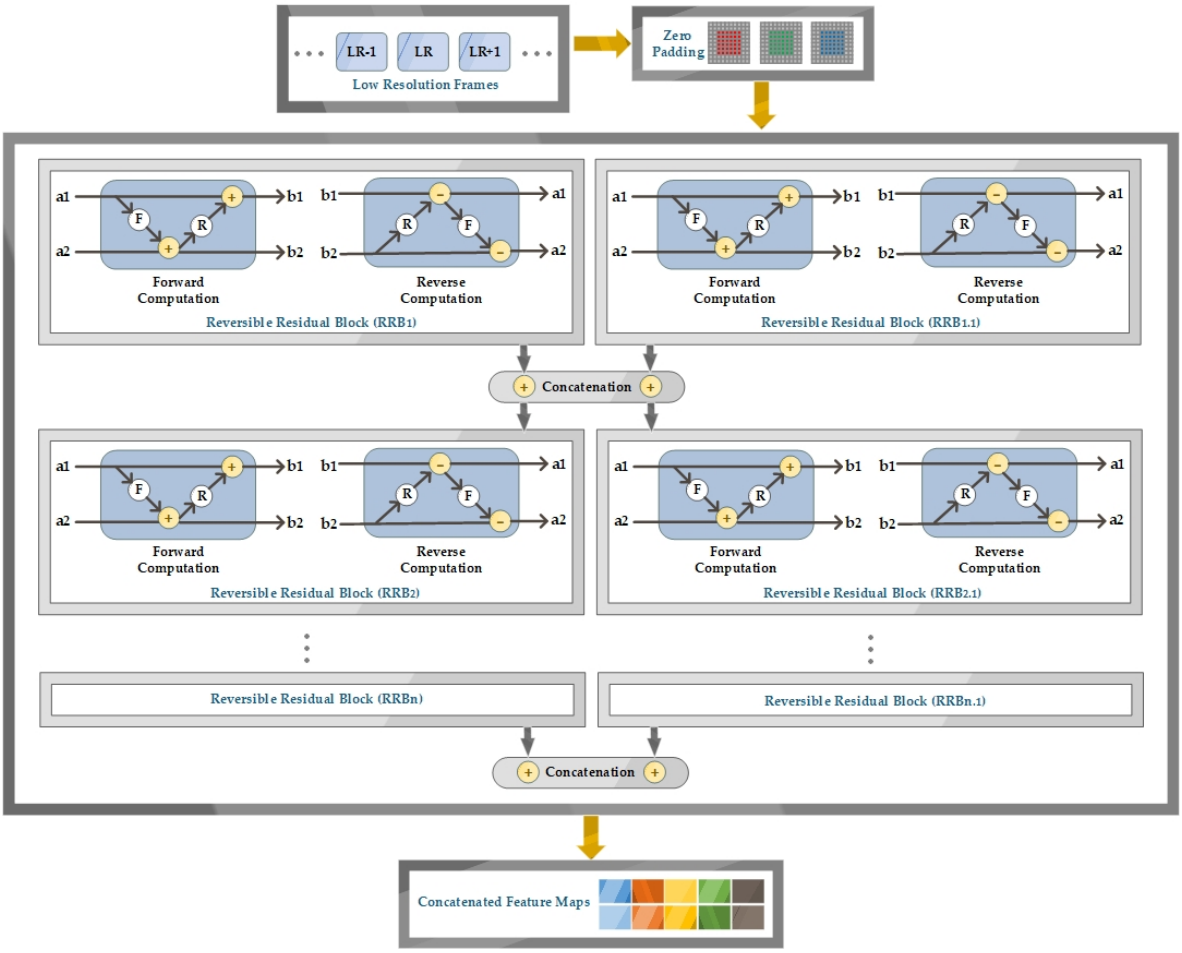

Figure 2. Architecture of the spatial generator.

4.3. Temporal Generator

The concatenated feature maps generated by the spatial generator are fed to the convolutional gated recurrent unit (ConvGRU) of the temporal generator. The ConvGRU retains the temporal information and consistency of each video frame. ConvGRU is computationally less expensive and acquires less memory than general long short term memory (LSTM). The convolutional GRU takes the linear GRU for computation and simply replaces the multiplication of matrix with convolutions. With x_t as input feature at time t, the ConvGRU [10] can be represented by the following equations (Equations (4)–(6)):

$$u_t = \sigma(W_u \star x_t + M_u \star h_{t-1} + b_u) \tag{4}$$

$$r_t = \sigma(W_r \star [u_t; x_t] + M_r \star h_{t-1} + b_u) \tag{5}$$

$$h_t = u_t \odot h_{t-1} + (1 - u_t) \odot \sigma(W_h \star [x_t; r_t \odot u_t] + M_h(r_t \odot h_{t-1}) + b_h \tag{6}$$

where u_t is an update gate at time t, r_t is a reset gate at time t and h_t is the update hidden state at time t. The σ is an activation function, W and M are the parameterized weight matrices and b is a vector. h_{t-1} is the hidden state from the previous time $t-1$. The $\star$ represents the convolutions in the GRU and the $\odot$ is the element-wise multiplication. The brackets $[\,]$ indicate concatenation of the features.

The concatenated feature maps from the spatial generator and the temporal features from the ConvGRU are then fed to the sparse matrix which contributes in selecting the required features and reduces the chances of overfitting. The fused features can be computed with the Equation (7).

$$F_{map} = F_{concat} \times M_S \tag{7}$$

where F_{concat} are the concatenated spatial and temporal features, M_S is the sparse matrix and $\times$ denotes matrix multiplication. Let the spatial feature maps denoted by F_S have c_1 channels and the temporal feature maps F_T have c_2 channels. Then the concatenated feature maps F_{concat} are defined as in Equation (8). where CF is the convolutional filter for temporal-spatial mapping and * is the convolutional operation with [,] being cross concatenation [17]

$$F_{concat} = [CF_{1\times1\times c_1\times c_2} \times F_T, F_S] \tag{8}$$

We use a deconvolutional layer to reconstruct the feature maps to high-resolution. The deconvolutional layer can be considered the upsampling layer that generates the high-resolution frames. We used 5×5 kernels and 512 feature maps for upsampling. Figure 3 shows the architectural view of the temporal generator.

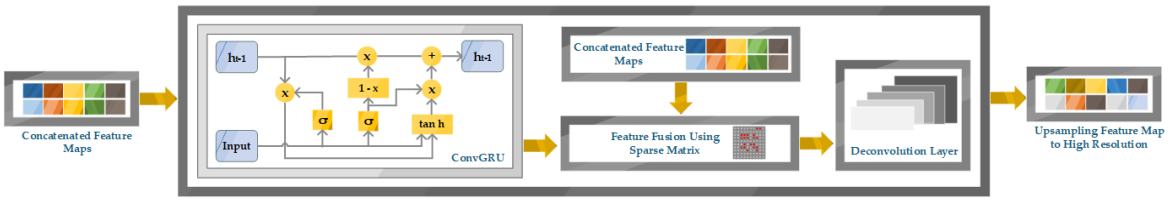

Figure 3. Architecture of the temporal generator.

4.4. Spatial Discriminator

The spatial discriminator receives the real data and the samples from the generator as input. The task of the spatial discriminator is to look at the individual frames and correctly identify whether the sample is from real data distribution or from the generator. The spatial discriminator uses a ResNet architecture. We use 3×3 convolutions, a batch normalization layer and a rectifier linear unit (ReLu) as the activation function. To change the number of strides we introduced a 1×1 convolutional layer. We used strided convolutions for down-sampling and the last layer did a binary classification to identify real or fake. The resolution of the frames depicts the number of residual blocks. The resulting latent vector l_v with skip connection was defined as 160 for resolution 512×512, 180 for 720×720 and 200 for 960×960. Figure 4 is an overview of the spatial discriminator.

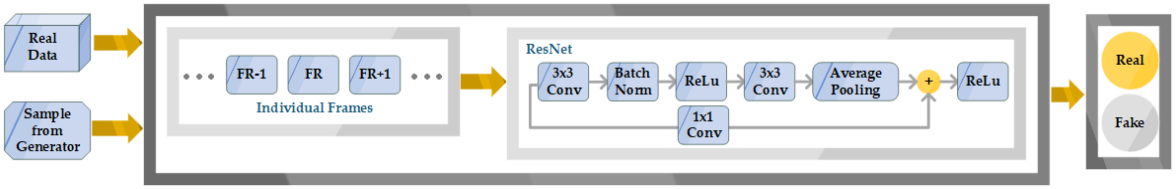

Figure 4. Architecture of the spatial discriminator.

4.5. Temporal Discriminator

Differently from [11], we use R(2 + 1)D convolutional ResNet instead of a 3D ResNet. We build (2 + 1)D block of N_i and 2D convolutional filters of $P_{i-1} \times 1 \times d \times d$, where P_i are temporal convolutions of size $N_i \times t \times 1 \times 1$. Here t denotes temporal features and d is the width and height of the spatial component. The advantages of using a (2 + 1)D ResNet is that the extra ReLU in each block of the 2D and 1D convolutions increases the number of non-linearities and the complexity of the functions

representing several small filters. Another advantage is that the optimization becomes easier when the 3D is broken down to (2 + 1)D in the ResNet architecture. We process the videos from the real or generated distribution using 2×2 average pooling. The discriminator is trained to differentiate between synthetic and real frames and to identify temporal features from each frame. The last layer is a binary classification layer the same as the spatial discriminator that outputs real or fake. Architecture for the temporal discriminator is shown in Figure 5.

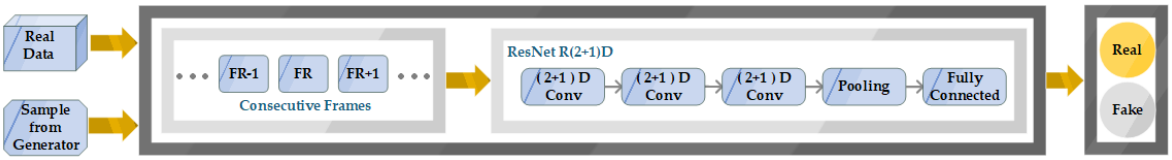

Figure 5. Architecture of the temporal discriminator.

5. Environment and Training

In this section, we discuss the environmental setup used for our experiment, the training details and the loss functions used to evaluate our model.

5.1. Experimental Setup

We have summarized the experimental setup in Table 1. We have used long term support version of Ubuntu 18.04.3 as the operating system. Total memory size was 32 GB. We used Python as our coding language.

Table 1. Components and specifications with versions of the system used for implementation.

Components	Specifications and Versions
Operating System	Ubuntu 18.04.3 LTS
Memory	32 GB
GPU	Nvidia GeForce GTX 1060 3 GB
API	Tensorflow 1.x
Interface	CUDA Toolkit version 10.1.243 and cuDNN version 7.6.5
CPU	Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz
Language	Python 3.6.5

5.2. Training

We trained and experimented with our model on the datasets Kinetics-700, UCF101, HMDB151 and IITH_Helmet2, as mentioned in the data section. Training was done with a scale factor of $4x$ from low-resolution to high-resolution frames. The mini-batch size was set to 16 and epoch of 2000. We trained both the generators and discriminators adversarially. We used Adam optimizer for optimization with $\beta 1 = 0.89$ and $\beta 2 = 0.9$. The model was trained until the generators and discriminators converged to the Nash equilibrium. If LR represents the low-resolution frames and HR represents the high-resolution frames, then the generator training loss can be computed as (9) [17]:

$$G_{LOSS} = \sum_{C=-k}^{k} \| HR_C - F(LR_C) \|_2^2 + \lambda \| M_S \|_1 \quad (9)$$

where k is the total number of frames that are consecutive, C represents the current frame, M_S is the sparse matrix, λ is the hyper-parameter and $\|_2$ represents the L2 norm.

5.3. Loss Functions

To measure the quality of the video reconstruction it is meaningful to compute the loss function. We evaluate the content loss L_{con}, adversarial loss L_{adv} and perceptual loss L_{perc} to reconstruct the high-resolution frames. Our total loss function is calculated as in (10):

$$L = L_{con} + \lambda_1 L_{adv} + \lambda_2 L_{perc} \tag{10}$$

where λ_1 and λ_2 are coefficients that help in balancing the total loss.

5.3.1. Content Loss

We have used the mean square error (MSE) to compute the model's content loss. This helps in retrieving the low-scaled information between the low-resolution frames and the reconstructed high-resolution frames. The error value corresponds to the pixel difference between the actual or ground-truth high-resolution frames and the generated high-resolution frames. Thus, evaluating the error can help to improve the accuracy of the reconstruction. The content loss can be defined as in (11) [37].

$$L_{con} = L_{MSE}(\theta) = \frac{1}{T_s} \sum_{i=1}^{T_s} \parallel HR_i - M(LR_i, \theta) \parallel^2 \tag{11}$$

where HR represents the real high-resolution frame, LR represents the low-resolution frames, M is the mapping function between low-resolution and high-resolution frames and i represents the total number of samples for training.

5.3.2. Adversarial Loss

Adversarial loss is computed for maximizing the probability of fooling the discriminator, by enhancing the performance of the generator to produce results exactly similar to the real data distribution. The adversarial loss penalizes the discriminator if it predicts incorrectly. Through the adversarial training both the generator and discriminator improve their performances and converge to the main goal. Adversarial loss can be defined as in (12) [37].

$$L_{adv} = -log(D_\psi(S \mid LR)) \tag{12}$$

where S is the synthetic frame and LR are the low resolution, distorted input frames.

5.3.3. Perceptual Loss

To enhance the features and textures of the video frames, perceptual loss is calculated. Perceptual loss concentrates on the fine features rather than the pixels; hence, it helps in reconstructing the low-resolution frames to high-resolution so that there will be maximum structural similarity in terms of features. In our proposed methodology we used the VGG-19 model [38] to extract features. Perceptual Loss is defined below in (13) [37].

$$L_{perc} = \frac{1}{W_n H_n} \sum_{i=1}^{W_n} \sum_{j=1}^{H_n} (\phi^n(HR)_{i,j} - \phi^n G(LR)_{i,j})^2 \tag{13}$$

where HR and LR are high and low-resolution frames, W_n and H_n are the width and height of the feature maps and ϕ represents the activation of a specific layer.

6. Evaluation

In this section we discuss the evaluation metrics used for performance measurement of our proposed model. We have computed the inception score (IS), Frechet inception distance (FID), peak

signal to noise ratio (PSNR) and structural similarity index (SSIM) to quantify the outcomes of our proposed model. As we can see from Table 2, our model performs the best in comparison to other existing models. We evaluated our model separately with three resolutions but the model can be used for any resolution. We used the already defined architectures of the GAN models mentioned in Table 2. In Table 2, CT is the computation time of per frame for every resolution in millisecond. PSNR measures the distortion in the video quality and SSIM measures the similarity between them. PSNR and SSIM can be calculated with the following equations (Equations (14) and (15)):

$$PSNR = 10 \times \log_{10} \frac{(PeakVal^2)}{MSE} \tag{14}$$

where MSE is mean squared error and $PeakVal$ is maximum resolution of the video frames.

$$SSIM(I, J) = \frac{(2_{\mu_I \mu_J} + C_1)(2\sigma_{IJ} + C_2)}{(\mu_{I^2} + \mu_{J^2} + C_1)(\sigma_{I^2} + \sigma_{J^2} + C_2)} \tag{15}$$

where I and J are two video frames; μ_I and μ_J represent the mean values; σ_I and σ_J represent the standard deviations; and σ_{IJ} is the covariance of frames I and J.

Table 2. Quantitative comparison of the proposed model.

Model	Resolution	IS	FID	PSNR	SSIM	CT-Per Frame (msec)
DVD-GAN	512p	27.24 ± 0.32	44.35 ± 0.45	23.26	83.90	750
	720p	29.71 ± 0.76	43.77 ± 0.53	23.47	80.12	800
	960p	24.32 ± 0.78	46.38 ± 0.21	24.35	81.34	900
BigGAN	512p	34.66 ± 0.74	38.57 ± 0.32	27.89	85.56	700
	720p	33.92 ± 0.25	38.22 ± 0.74	27.63	86.73	750
	960p	35.43 ± 0.66	35.84 ± 0.45	28.93	86.98	800
BigGAN-deep	512p	33.45 ± 0.48	29.99 ± 0.74	31.32	89.99	700
	720p	33.69 ± 0.44	28.43 ± 0.87	32.75	89.92	780
	960p	36.43 ± 0.88	27.65 ± 0.35	32.66	90.12	850
Proposed GAN	512p	48.44 ± 0.36	17.43 ± 0.63	40.12	93.45	500
	720p	49.65 ± 0.83	16.94 ± 0.39	41.32	93.98	650
	960p	49.99 ± 0.64	16.92 ± 0.55	41.59	94.25	700

Figure 6 shows the input low-resolution videos and the high-resolution output using our proposed GAN model. In Figures 7 and 8 we show the raw low-resolution frame samples from Kinetic-700, UCF101 and HMDB51 datasets. Figures 9 and 10 show the generated high-resolution output of the raw samples using our proposed GAN.

Median recover error (MRE) and MRE-gap described in the paper [39] have been used to detect whether our model overfits. In [39], the author states that the p-value of the Kolmogorov–Smirnov test (KS) can be evaluated to estimate the degree of overfitting. A model with a p-value that is below 1% and MRE-gap above 8% has been considered to be overfitting. Our model has generated results for p-value above 1% and MRE-gap below 8% which concludes that our model prevents overfitting. Figure 11 shows the loss over time for training and validation.

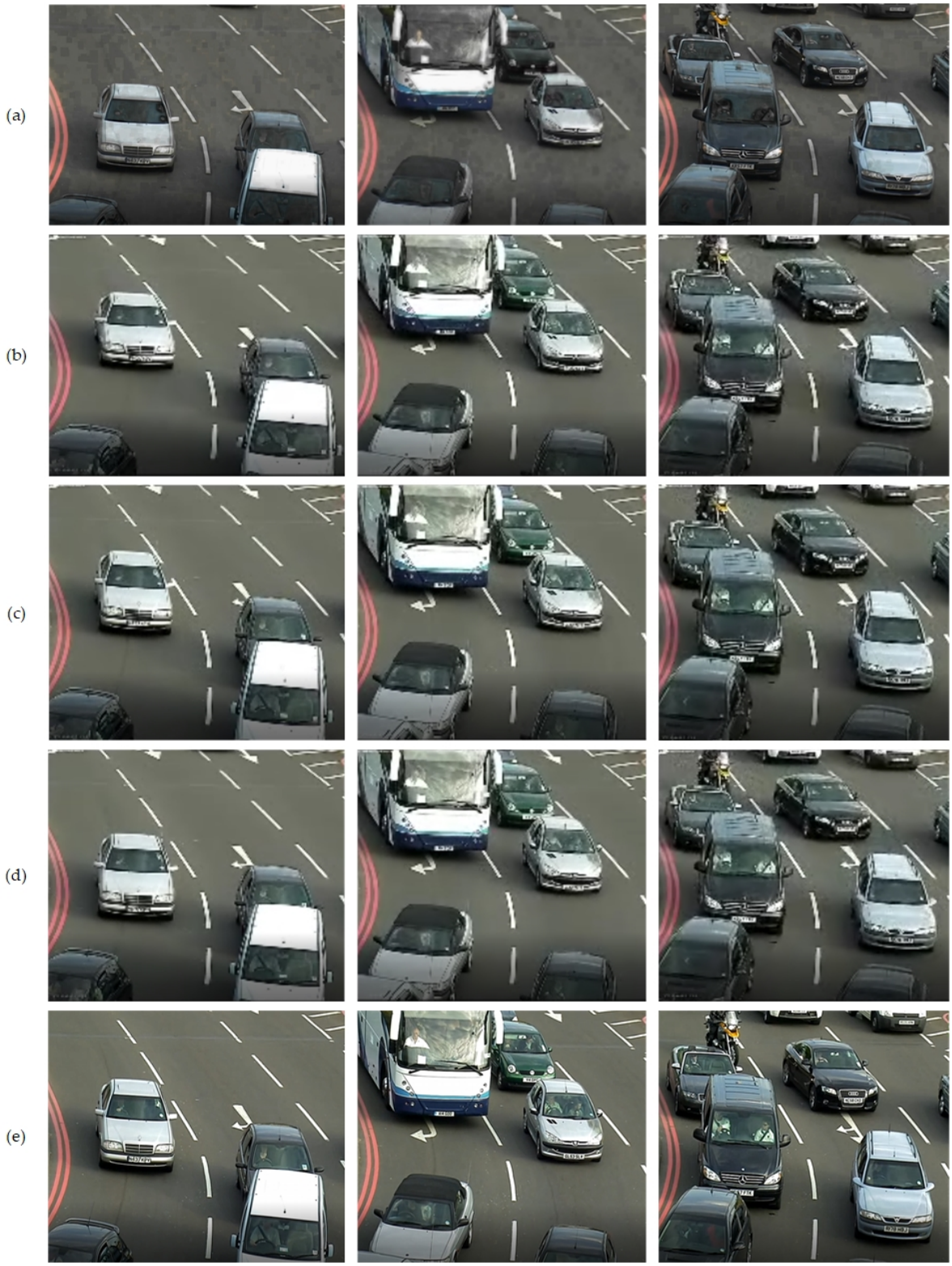

Figure 6. Input: (**a**) low-resolution; output of high-resolution video samples from IITH_Helmet2 dataset using: (**b**) DVD-GAN, (**c**) BigGAN, (**d**), BigGAN-deep and (**e**) proposed GAN.

Figure 7. A random batch of raw samples from Kinetics-700, UCF101 and HMDB51 datasets trained on 12 frames.

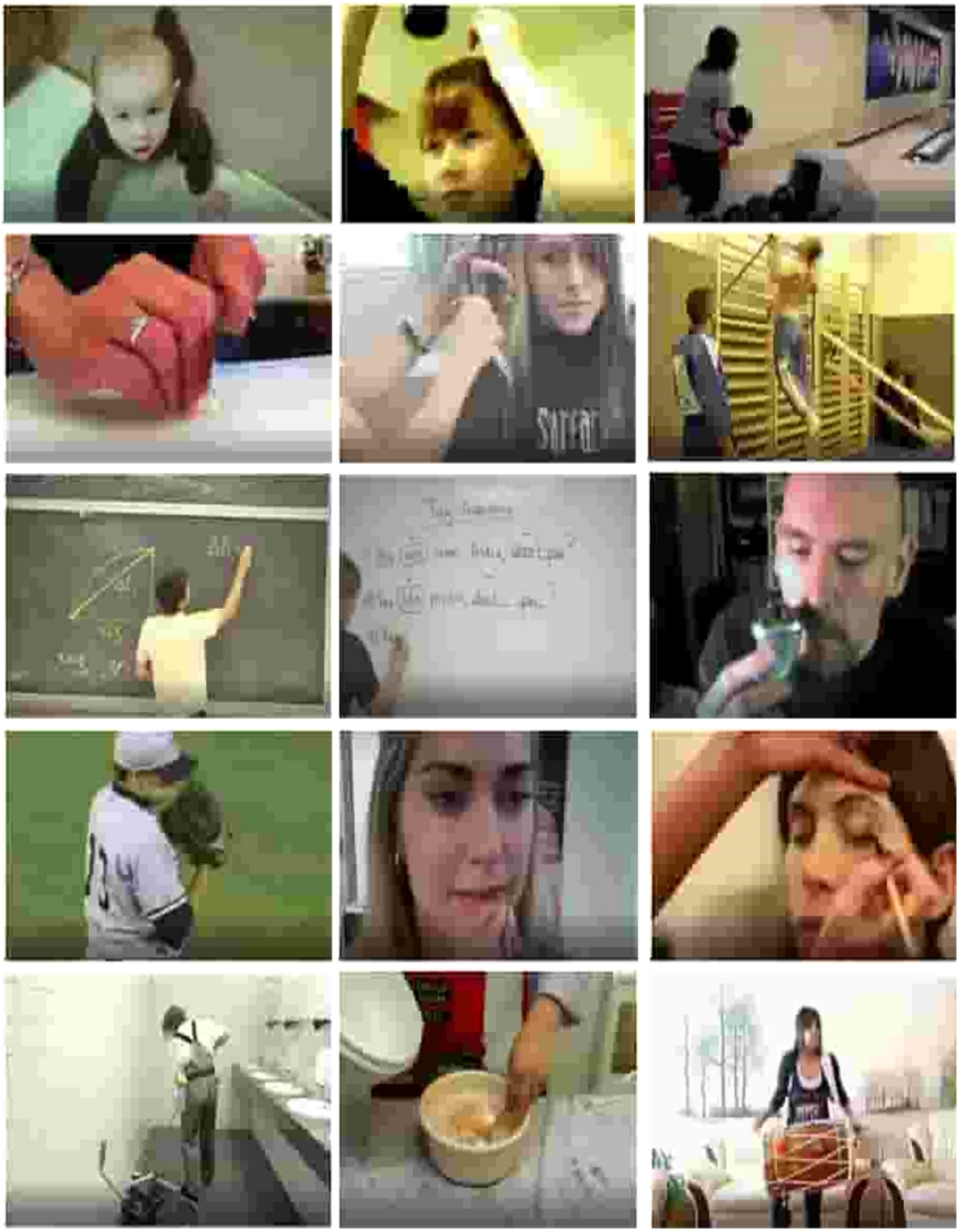

Figure 8. A random batch of raw samples from Kinetics-700, UCF101 and HMDB51 datasets trained on 48 frames.

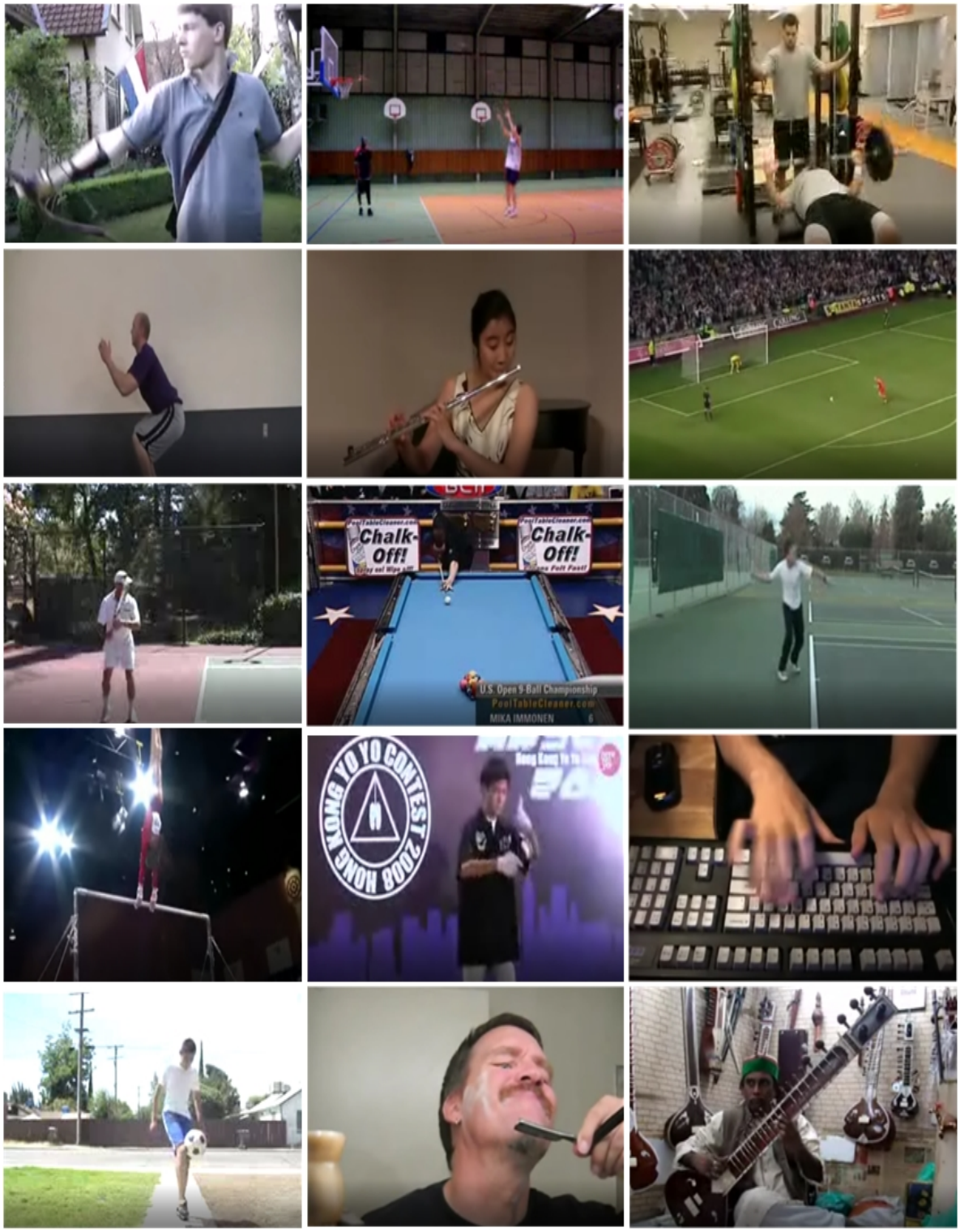

Figure 9. Generated high-resolution frames of the raw samples.

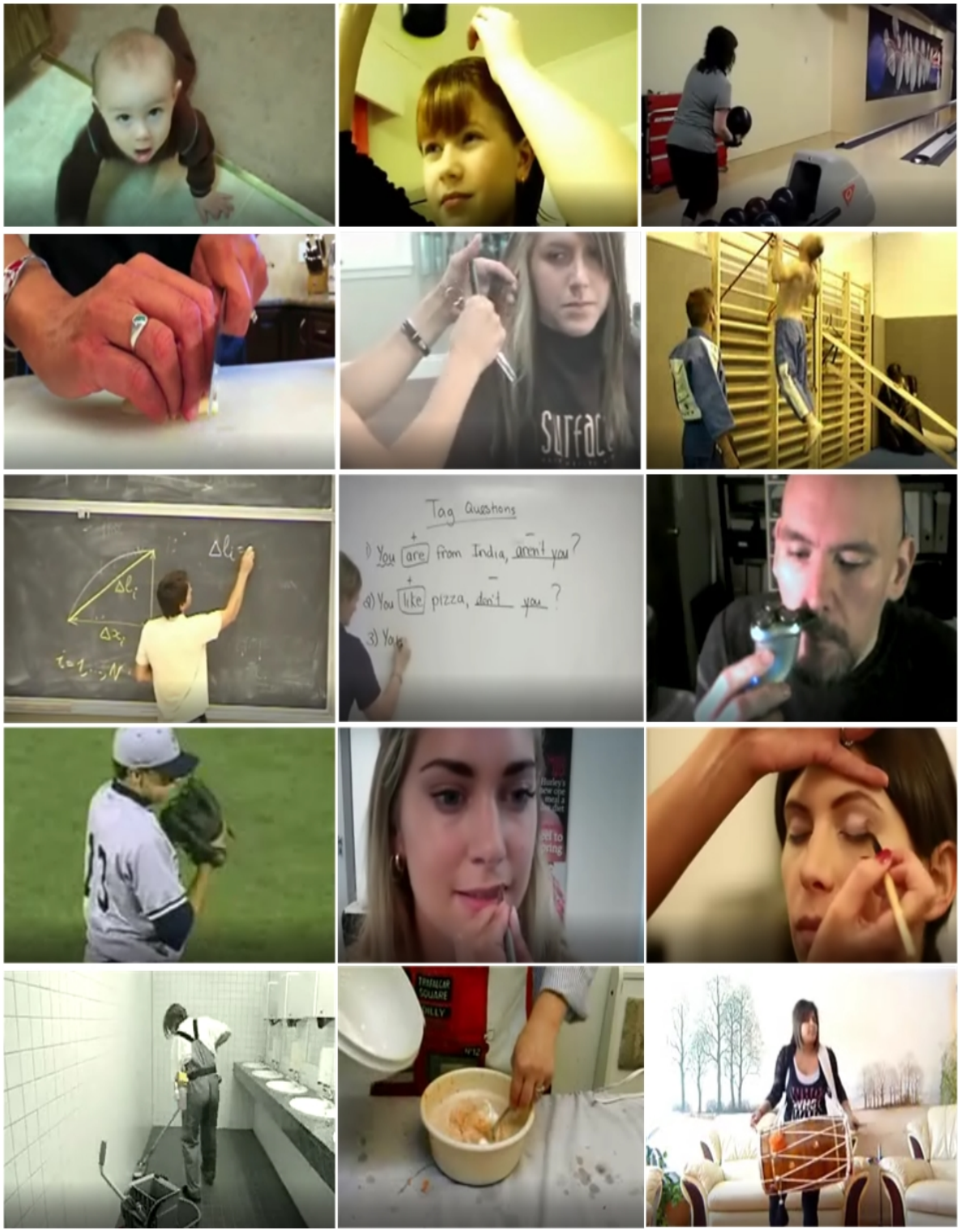

Figure 10. Generated high-resolution frames of the raw samples.

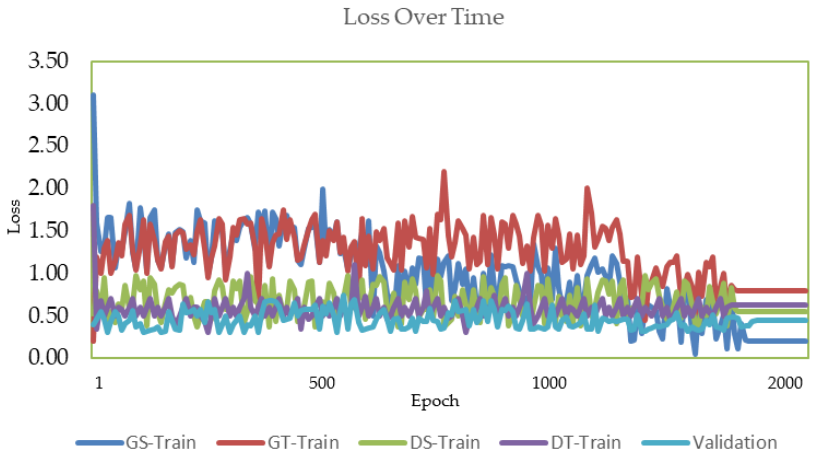

Figure 11. Loss over time for training and validation.

7. Conclusions

This paper presents a generative adversarial network-based model for real-time super-resolution of low-resolution CCTV videos. Quantitative and qualitative results show that our approach requires less computational overhead and produces better results than existing GAN models for video super-resolution. We have considered spatio-temporal features while developing our model for both the generators and discriminators. This helped us to extract intricate features while considering foreground and background motion. We have experimented with our model on various datasets, including the Kinetics-700, UCF101, HMDB51 and IITH_Helmet2 video datasets. Our model is scalable and we have trained our model on large-scale datasets. In future research work, we would like to consider the space issue for CCTV videos and combine our proposed model with technology that could produce high-resolution videos along with the capacity of consuming less memory.

Author Contributions: Conceptualization, D.H.; formal analysis, D.H.; funding acquisition, Y.-C.B.; methodology, D.H.; writing—review and editing, D.H.; investigation, Y.-C.B.; resources, Y.-C.B.; project administration, Y.-C.B.; supervision, Y.-C.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government (MOTIE) (N0002327, The Establishment Project of Industry-University Fusion District).

Conflicts of Interest: The authors declare no conflict of interest regarding the design of this study, analyses or writing of this manuscript.

References

1. Ashby, M.P. The value of CCTV surveillance cameras as an investigative tool: An empirical analysis. *Eur. J. Crim. Policy Res.* **2017**, *23*, 441–459. [CrossRef]
2. International Trends in Video Surveillancepublic Transport Gets Smarter. Available online: https://www.uitp.org/sites/default/files/cck-focus-papers-files/1809-Statistics%20Brief%20-%20Videosurveillance-Final.pdf (accessed on 15 July 2020).
3. Size of the Global Video Surveillance Market between 2016 and 2025. Available online: https://www.statista.com/statistics/864838/video-surveillance-market-size-worldwide/ (accessed on 15 July 2020).
4. Khan, P.W.; Byun, Y. A Blockchain-Based Secure Image Encryption Scheme for the Industrial Internet of Things. *Entropy* **2020**, *22*, 175. [CrossRef]
5. Park, N.; Kim, B.G.; Kim, J. A Mechanism of Masking Identification Information regarding Moving Objects Recorded on Visual Surveillance Systems by Differentially Implementing Access Permission. *Electronics* **2019**, *8*, 735. [CrossRef]
6. Khan, P.W.; Xu, G.; Latif, M.A.; Abbas, K.; Yasin, A. UAV's agricultural image segmentation predicated by clifford geometric algebra. *IEEE Access* **2019**, *7*, 38442–38450. [CrossRef]

7. Clark, A.; Donahue, J.; Simonyan, K. Efficient video generation on complex datasets. *arXiv* **2019**, arXiv:1907.06571.
8. Khan, P.W.; Byun, Y.C.; Park, N. A Data Verification System for CCTV Surveillance Cameras Using Blockchain Technology in Smart Cities. *Electronics* **2020**, *9*, 484. [CrossRef]
9. Yang, W.; Feng, J.; Xie, G.; Liu, J.; Guo, Z.; Yan, S. Video super-resolution based on spatial-temporal recurrent residual networks. *Comput. Vis. Image Underst.* **2018**, *168*, 79–92. [CrossRef]
10. Ballas, N.; Yao, L.; Pal, C.; Courville, A. Delving deeper into convolutional networks for learning video representations. *arXiv* **2015**, arXiv:1511.06432.
11. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6450–6459.
12. Tulyakov, S.; Liu, M.Y.; Yang, X.; Kautz, J. Mocogan: Decomposing motion and content for video generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1526–1535.
13. Saito, M.; Saito, S.; Koyama, M.; Kobayashi, S. Train Sparsely, Generate Densely: Memory-Efficient Unsupervised Training of High-Resolution Temporal GAN. *Int. J. Comput. Vis.* **2020**. [CrossRef]
14. Saito, M.; Matsumoto, E.; Saito, S. Temporal generative adversarial nets with singular value clipping. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2830–2839.
15. Brock, A.; Donahue, J.; Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv* **2018**, arXiv:1809.11096.
16. Gomez, A.N.; Ren, M.; Urtasun, R.; Grosse, R.B. The reversible residual network: Backpropagation without storing activations. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 2214–2224.
17. Zhu, X.; Li, Z.; Zhang, X.Y.; Li, C.; Liu, Y.; Xue, Z. Residual invertible spatio-temporal network for video super-resolution. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 5981–5988.
18. Samaniego, E.; Anitescu, C.; Goswami, S.; Nguyen-Thanh, V.M.; Guo, H.; Hamdia, K.; Zhuang, X.; Rabczuk, T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Comput. Methods Appl. Mech. Eng.* **2020**, *362*, 112790. [CrossRef]
19. Guo, H.; Zhuang, X.; Rabczuk, T. A deep collocation method for the bending analysis of Kirchhoff plate. *Comput. Mater Contin.* **2019**, *59*, 433–456. [CrossRef]
20. Vondrick, C.; Pirsiavash, H.; Torralba, A. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2016; pp. 613–621.
21. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef] [PubMed]
22. Kim, J.; Kwon Lee, J.; Mu Lee, K. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
23. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
24. Lai, W.S.; Huang, J.B.; Ahuja, N.; Yang, M.H. Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 624–632.
25. Tai, Y.; Yang, J.; Liu, X. Image super-resolution via deep recursive residual network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3147–3155.
26. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 694–711.

27. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2014; pp. 2672–2680.
28. Sajjadi, M.S.; Scholkopf, B.; EnhanceNet, M.H. Single Image Super-Resolution through Automated Texture Synthesis. *Max-Planck-Inst. Intell. Syst. Spemanstr* 2016, *23*, 4501–4510.
29. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 22–25 July 2017; pp. 136–144.
30. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
31. Carreira, J.; Noland, E.; Hillier, C.; Zisserman, A. A short note on the kinetics-700 human action dataset. *arXiv* **2019**, arXiv:1907.06987.
32. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
33. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2556–2563.
34. Vishnu, C.; Singh, D.; Mohan, C.K.; Babu, S. Detection of motorcyclists without helmet in videos using convolutional neural network. In Proceedings of the 2017 International Joint Conference on Neural Networks, (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 3036–3041. [CrossRef]
35. Dinh, L.; Krueger, D.; Bengio, Y. Nice: Non-linear independent components estimation. *arXiv* **2014**, arXiv:1410.8516.
36. Dinh, L.; Sohl-Dickstein, J.; Bengio, S. Density estimation using real nvp. *arXiv* **2016**, arXiv:1605.08803.
37. Jiang, Y.; Li, J. Generative Adversarial Network for Image Super-Resolution Combining Texture Loss. *Appl. Sci.* **2020**, *10*, 1729. [CrossRef]
38. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
39. Webster, R.; Rabin, J.; Simon, L.; Jurie, F. Detecting overfitting of deep generative networks via latent recovery. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–21 June 2019; pp. 11273–11282.